

EXAM P2P 2024-2025

Médiathèque Digitale (MD) en P2P

Conception de l'architecture

L'architecture pourra être vue de manière modulaire, divisée en plusieurs composants clés :

- **Portail Web (www.md.eu)**
 - **Frontend** : L'interface utilisateur où les usagers peuvent s'inscrire, rechercher des contenus, télécharger, et interagir avec la communauté.
 - **Backend** : Gère les requêtes des utilisateurs, les bases de données, et la logique d'accès aux contenus. Il s'agit d'un serveur centralisé, mais ce backend peut être minimaliste grâce à l'utilisation de DHT pour gérer les données de manière décentralisée.
- **App mobile (myMD)**
 - **Frontend mobile** : Une version plus légère de l'interface qui permet l'accès à toutes les fonctions principales sur des dispositifs mobiles.
 - **Backend mobile** : Communication avec le backend central via API RESTful pour les actions principales comme la gestion des profils et des contenus.
- **Système de stockage décentralisé**
 - **Nodes P2P** : Chaque usager possède un dispositif (fixe ou mobile) qui agit comme un nœud dans le réseau P2P. Ces dispositifs gèrent l'upload, le stockage, et le téléchargement de contenus.
 - **DHT (Chord/Kademlia)** : Utilisé pour indexer les contenus de manière décentralisée, permettant aux usagers de trouver facilement les fichiers qu'ils recherchent, sans dépendre d'un serveur centralisé.
- **Système de publication de contenus**
 - **Bittorrent, Magnet Links et Trackers** : L'utilisateur peut publier ses contenus en les liant à des "Bittorrent Magnet links" qui référencent les fichiers dans le réseau P2P. Un tracker centralisé (type PirateBay) ou une DHT peut être utilisé pour trouver et distribuer ces liens.

TODO : Description des API pour l'App myMD et le portail Web

a) Inscription à la MD (via API)

Méthode : POST

Explication : Cette méthode permettra à un nouvel utilisateur de s'inscrire à la MD. Le système enregistrera un nouveau profil utilisateur dans le réseau P2P. La base de données P2P contiendra ces informations qui peuvent être modifiées par l'utilisateur à tout moment.

b) Upload de contenu (via API)

Méthode : POST

Explication : Lorsque l'utilisateur voudra uploader un fichier, ce dernier sera découpé en morceaux, puis distribué sur le réseau P2P. Les métadonnées, telles que le nom du fichier et l'ID utilisateur, seront enregistrées pour la gestion future de la recherche.

c) Recherche de contenu (via API)

Méthode : GET

Explication : Cette méthode interrogera le réseau DHT pour retrouver les fichiers correspondant à la requête. Une fois le contenu trouvé, un lien Bittorrent sera retourné à l'utilisateur pour télécharger le fichier.

Conception du mécanisme de Ledger avec Blockchain

La blockchain pourra être utilisée pour enregistrer chaque action de l'utilisateur : l'inscription, l'upload de contenu, les téléchargements, etc. Cela permettra de suivre les crédits et les points d'un utilisateur, d'assurer la transparence des opérations, et d'instaurer un système de récompenses ou de validation des actions.

TODO : Description des API pour la blockchain et du mécanisme de ledger

- **Blockchain** : Chaque transaction (upload, download, inscription) est un bloc.
- **Tokenisation** : Chaque action génère des "tokens" que l'utilisateur peut utiliser pour obtenir des crédits ou des services supplémentaires dans la MD.

Cela pourra également être intégré avec des mécanismes de paiement pour la monétisation des échanges.

Monétisation des échanges avec un système de paiement électronique

Pour rendre l'échange de contenus monétisable, on pourra introduire un système de paiement *receive-per-upload* et *pay-per-download*. Les usagers devront payer des crédits ou des tokens pour télécharger des fichiers ou recevoir des crédits ou des tokens pour offrir une disponibilité des fichiers.

TODO : Description des API pour la monétisation des échanges

- **Exemple de paiement** :
 - **Upload** : L'utilisateur reçoit un certain nombre de crédits lorsqu'il télécharge un fichier.
 - **Download** : L'utilisateur doit payer un certain nombre de crédits pour télécharger un fichier.

Les paiements peuvent être gérés par une API de paiement électronique, par exemple via des services comme PayPal, ou même un portefeuille virtuel interne à la plateforme.

Garantir l'anonymat des communications

Afin de protéger la vie privée des usagers et garantir l'anonymat, plusieurs mécanismes pourront être envisagés :

TODO : Description des API pour la garantir l'anonymat des échanges

- **Cryptage des communications** : Toutes les communications, telles que les échanges de fichiers et les requêtes, doivent être cryptées à l'aide de protocoles comme SSL/TLS.
- **Anonymat de l'utilisateur** : En plus du cryptage, un réseau P2P peut également masquer l'adresse IP de l'utilisateur en utilisant des techniques de "mix network" ou de "Tor-like networks" pour éviter que des tiers ne traquent l'activité des utilisateurs.

[FACULTATIF] Interconnexion de plusieurs médiathèques digitales

Plusieurs communautés souhaitent partager leurs bibliothèques sans fusionner leurs bases de données, Ceci est un défi intéressant qui implique de maintenir l'autonomie de chaque MD tout en permettant un échange d'information fluide et sécurisé. Cette interconnexion peut être réalisée via un mécanisme de communication entre réseaux P2P sans violer la séparation entre chaque MD.

TODO : Description des API pour la garantir l'interconnexion des échanges entre MD

a. Contexte du problème :

Chaque village possède une base de données distribuée, et chaque base de données utilise un système de hachage distribué (DHT) tel que **Chord** ou **Kademlia**. Le but ici est de permettre à ces différentes communautés de se partager du contenu (documents, vidéos, livres, etc.) sans que les bases de données elles-mêmes ne soient fusionnées ou centralisées.

b. Objectif de l'interconnexion :

Les différentes médiathèques digitales échangeront des informations ses contenus (par exemple, les titres de livres ou vidéos) tout en préservant leur indépendance. Cela nécessitera une approche **synaptique à la Liquori et al.** où chaque MD pourra interagir avec d'autres sans compromettre la sécurité, l'intégrité ou la confidentialité de ses données locales.

c. Proposition d'architecture d'interconnexion :

Pour réaliser cette interconnexion, utiliser des **"Synapses" à la Liquori et al**, un mécanisme qui agit comme un connecteur entre plusieurs DHTs distincts sans avoir besoin de fusionner les bases de données. L'usage de **synapses** entre différentes MDs permettra d'établir un **réseau décentralisé** et interconnecté sans avoir à fusionner les bases de données de chaque MD. Chaque MD conservera son autonomie tout en permettant aux usagers de profiter d'un accès plus large aux contenus disponibles dans d'autres MDs ainsi qu'une résilience aux pannes. Avantages :

- **Indépendance des MDs** : Chaque communauté peut maintenir ses propres données locales et ses politiques tout en accédant aux métadonnées des autres MDs.
- **Scalabilité** : L'architecture permet d'ajouter de nouvelles MDs sans fusionner les bases de données existantes, ce qui offre une scalabilité et une flexibilité importante.
- **Anonymat et sécurité** : Les synapses ne stockent que des métadonnées (liens et informations de contenu) et non des données personnelles ou des fichiers eux-mêmes, ce qui garantit un meilleur contrôle de la confidentialité.
- **Efficacité** : Les MDs peuvent interagir efficacement via les synapses sans avoir besoin de synchroniser des volumes massifs de données.